Solving PhaseLift by low-rank Riemannian optimization methods for complex semidefinite constraints

Wen Huang
Université Catholique de Louvain

Kyle A. Gallivan  Xiangxiong Zhang
Florida State University  Purdue University

March 2015
The Phase Retrieval problem concerns recovering a signal given the modulus of its linear transform;

It is important in many applications, e.g.,
- X-ray crystallography imaging [Har93];
- Diffraction imaging [BDP⁺07];
- Optics [Wal63];
- Microscopy [MISE08];

The Fourier transform is considered;
Problem Statement

Recover the signal $x : \mathbb{R}^s \rightarrow \mathbb{C}$ from intensity measurements of its Fourier transform, $|\tilde{x}(u)| = \left| \int_{\mathbb{R}^s} x(t) \exp(-2\pi u \cdot t \sqrt{-1}) dt \right|$;

Discrete form

find $x \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_s}$, s. t. $|Ax| = b$,

where $x = \text{vec}(x) \in \mathbb{R}^n$, $n = n_1 n_2 \cdots n_s$ and $A \in \mathbb{C}^{m \times n}$ defines the Discrete Fourier transform;
Solution of the discrete form may be not unique.

Oversampling in the Fourier domain is a standard method to obtain a unique solution.
  - No benefit for most 1D signals, see e.g., [San85].
  - Give a unique solution for multiple dimensional problems for constrained signals, see e.g. [BS79, Hay82, San85].
  - Algorithms based on alternating projection are used.
PhaseLift [CESV13] and PhaseCut [WDM13]: Combining using multiple structured illuminations or masks with convex programming;
A unique rank one solution up to a global phase factor [CESV13, CL13, CSV13, WDM13];
Stability [CL13, CSV13, WDM13];
Convex programming solvers, e.g., SDPT3 [TTT99] or TFOCS [BCG11];
The PhaseLift framework is considered in this presentation.
The known illumination fields on the discrete signal domain
\( \mathbf{w}_r \in \mathbb{C}^{n_1 \times n_2 \times \cdots n_s}, \ r = 1, \ldots \ell. \)

Let \( \mathbf{w}_r \) denote \( \text{vec}(\mathbf{w}_r) \). One illumination field gives an equation

\[
|A \text{Diag}(\mathbf{w}_r)x| = b_r
\]

where \( \text{Diag}(\mathbf{w}_r) \) denotes an \( n \)-by-\( n \) diagonal matrix the diagonal entries of which are \( \mathbf{w}_r \).

\( \ell \) fields yields

\[
\begin{bmatrix}
A & \text{Diag}(\mathbf{w}_1) \\
A & \text{Diag}(\mathbf{w}_2) \\
\vdots & \vdots \\\nA & \text{Diag}(\mathbf{w}_\ell)
\end{bmatrix}
\begin{bmatrix}
x
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_\ell
\end{bmatrix}
:= b
Therefore, the linear operator $A$ for PhaseLift using the Fourier transform, denote $Z$, is

$$Z = \begin{pmatrix}
    A & & \\
    & A & \\
    & & \ddots & A \\
\end{pmatrix}
\begin{pmatrix}
    \text{Diag}(w_1) \\
    \text{Diag}(w_2) \\
    \vdots \\
    \text{Diag}(w_\ell)
\end{pmatrix}$$

PhaseLift problem:

find $x \in \mathbb{C}^n$, 

s. t. $|Zx| = b$, or equivalently, $\text{diag}(Zxx^* Z^*) = b^2$,

where $*$ denotes the conjugate transpose operator.
Let $X \in \mathbb{C}^{n \times n}$ denote $xx^*$. The Phase Retrieval problem becomes

$$\text{find } X, \quad \text{s. t. } \text{diag}(ZXZ^*) = b^2, X \geq 0 \text{ and } \text{rank}(X) = 1,$$

or equivalently

$$\min \text{rank}(X), \quad \text{s. t. } \text{diag}(ZXZ^*) = b^2, \text{ and } X \geq 0,$$

where $X \geq 0$ denotes $X$ is Hermitian positive semidefinite.

Convex programming

$$\min \text{trace}(X), \quad \text{s. t. } \text{diag}(ZXZ^*) = b^2, \text{ and } X \geq 0.$$
Measurements with noise, $b^2 \in \mathbb{R}^m$, are sampled from a probability distribution $p(\cdot; \mu)$, where $\mu = \text{diag}(ZXZ^*)$.

Minimize the negative log-likelihood function

$$\min_{\mu} - \log(p(b; \mu))$$

such that $\mu = \text{diag}(Zxx^*Z^*)$,

Similarly, an alternate problem can be used:

$$\min - \log(p(b; \mu)) + \kappa \text{trace}(X), \quad \text{s. t.} \quad \text{diag}(ZXZ^*) = b^2, \quad \text{and} \quad X \geq 0.$$ 

where $\kappa$ is a positive constant.

If the likelihood is log-concave, then it is a convex problem, e.g., for Poisson or Gaussian distributions.
PhaseLift: Nonconvex Approach

- The complexity can be too high in convex approach.
- The alternate problems are
  
  \[
  \begin{align*}
  \text{noiseless: } & \min_{X \geq 0} \|b^2 - \text{diag}(ZXZ^*)\|_2^2 + \kappa \text{trace}(X), \\
  \text{noise: } & \min_{X \geq 0} -\log(p(b; \text{diag}(ZXZ^*))) + \kappa \text{trace}(X)
  \end{align*}
  \]

  where \(\kappa\) is a positive constant;
- They are used in [CESV13] and reweighting is used to promote low-rank solutions;
- This motivates us to consider the optimization problem

  \[
  \min_{X \geq 0} H(X)
  \]

  and the desired minimizer is low rank. In particular for the PhaseLift problem, the rank of desired minimizer is 1.
Suppose the rank of desired minimizer $r^*$ is at most $p$.

The domain $\{X \in \mathbb{C}^{n \times n} | X \geq 0\}$ can be replaced by $\mathcal{D}_p$, where $\mathcal{D}_p = \{X \in \mathbb{C}^{n \times n} | X \geq 0, \text{rank}(X) \leq p\}$.

An alternate cost function can be used:

$$F_p : \mathbb{C}^{n \times p} \rightarrow \mathbb{R} : Y_p \mapsto H(Y_p Y_p^*).$$

Note that for the PhaseLift problem, choosing $p = 1$ is equivalent not to do the Lifting step. Choosing $p > 1$ yields computational and theoretical benefits.

This idea is not new and has been discussed in [BM03] and [JBAS10] for real positive semidefinite matrix constraints.
Theorem

If \( Y_p^* \in \mathbb{C}^{n \times p} \) is a rank deficient minimizer of \( F_p \), then \( Y_p Y_p^* \) is a stationary point of \( H \).

In addition, if \( H \) is a convex cost function, \( Y_p Y_p^* \) is a global minimizer of \( H \).

- The real version of the optimality condition is given in [JBAS10].
Equivalence: if $Y_p Y_p^* = \tilde{Y}_p \tilde{Y}_p^*$, then $F_p(Y_p) = F_p(\tilde{Y}_p)$;

Quotient manifolds are used to remove the equivalence:

- Equivalent class of $Y_r \in \mathbb{C}_{n \times r}$ is $[Y_r] = \{Y_r O_r | O_r \in \mathcal{O}_r\}$, where $1 \leq r \leq p$, $\mathbb{C}_{n \times r}$ denotes the $n$-by-$r$ complex noncompact Stiefel manifold and $\mathcal{O}_r$ denote the $r$-by-$r$ complex rotation group;
- A fixed rank quotient manifold $\mathbb{C}_{n \times r}/\mathcal{O}_r = \{[Y_r] | Y_r \in \mathbb{C}_{n \times r}\}$, $1 \leq r \leq p$;

Function on a fixed rank manifold is

$$f_r : \mathbb{C}_{* \times r}/\mathcal{O}_r \rightarrow \mathbb{R} : [Y_r] \mapsto F_r(Y_r) = H(Y_r Y_r^*);$$

- Optimize the cost function $f_r$ and update $r$ if necessary;
- A similar approach is used in [JBAS10] for real problems;
Update Rank Strategy

- Most of work is to choose a upper bound $k$ for the rank and optimize over $\mathbb{C}^{n \times k}$ or $\mathbb{R}^{n \times k}$.
- Increasing rank by a constant [JBAS10, UV14]
  - Descent
  - Globally converge
- Dynamically search for a suitable rank [ZHGG+15]
  - Not descent
  - Globally converge
Rank reduce is used for the problem in PhaseLift;
The rank is reduced if the singular values of an iterate have notable bias:
- Suppose $r$ is the rank of current iterate and $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_r$ are its singular values;
- Given a threshold $\delta \in (0, 1)$, the next rank is $q$ if $\sigma_q > \delta \tilde{\sigma}$ and $\sigma_{q+1} \leq \delta \tilde{\sigma}$, where $\tilde{\sigma} = \|\text{Diag}(\sigma_1, \ldots, \sigma_r)\|_F / \sqrt{r}$;
The next iterate is given by truncating relative small singular values;
Riemannian optimization algorithms are used to optimize the problem on the fixed rank manifold $\mathbb{C}^{n \times r}_* / O_r$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line search Newton (RNewton)</td>
<td>[AMS08]</td>
</tr>
<tr>
<td>Trust region Newton (RTR-Newton)</td>
<td>[Bak08]</td>
</tr>
<tr>
<td>BFGS (RBFGS)</td>
<td>[RW12, HGA14]</td>
</tr>
<tr>
<td>Limited memory version of BFGS (LRBFGS)</td>
<td>[HGA14]</td>
</tr>
<tr>
<td>Trust region symmetric rank one update method (RTR-SR1)</td>
<td>[HAG14]</td>
</tr>
<tr>
<td>Limited memory version of RTR-SR1 (LRTR-SR1)</td>
<td>[HAG14]</td>
</tr>
<tr>
<td>Riemannian conjugate gradient method (RCG)</td>
<td>[AMS08]</td>
</tr>
</tbody>
</table>
Algorithm 1

1: Set initial rank \( r = p \);
2: \textbf{for} \( k = 0, 1, 2, \ldots \) \textbf{do}
3: \hspace{1em} Apply Riemannian method for cost function \( f_r \) over \( \mathbb{C}^{n \times r} / \mathcal{O}_r \) with initial point \([Y_r^{(k)}]\) until \( i \)-th iterate \([W^{(i)}]\) satisfying \( \| \text{grad} f_r([W^{(i)}]) \| < \epsilon \) or the requirement of reducing rank with threshold \( \delta \);
4: \hspace{1em} \textbf{if} \( \| \text{grad} f_r([W^{(i)}]) \| < \epsilon \) \textbf{then}
5: \hspace{2em} Find a minimizer \([W] = [W^{(i)}]\) over \( \mathbb{C}^{n \times p} / \mathcal{O}_p \) and return;
6: \hspace{1em} \textbf{else} \{iterate in the Riemannian optimization method meets the requirements of reducing rank\}
7: \hspace{2em} Reduce the rank to \( q < r \) based on truncation with threshold \( \delta \) and obtain an output \( \hat{W} \in \mathbb{C}^{n \times q} \);
8: \hspace{1em} \( r \leftarrow q \) and set \([Y_r^{(k+1)}] = [\hat{W}]\);
9: \hspace{1em} \textbf{end if}
10: \textbf{end for}
Numerical Experiments

Artificial Data sets

- The entries of true solution $x_*$ and the masks $w_i, i = 1, \ldots, l$ are drawn from the standard normal distribution;
- $x_*$ is further normalized by $\|x_*\|_2$;
- $w_i, i = 1, \ldots, l$ is further normalized by $\sqrt{n}$;
- The measurements $b^2$ is set to be $\text{diag}(Zx_*x_*^*Z^*) + \epsilon$, where the entries of $\epsilon \in \mathbb{R}^m$ are drawn from the normal distribution with mean 0 and variance $\tau$. 
The cost function in this case is

\[ f_r([Y_r]) = \frac{\|b^2 - \text{diag}(ZY_r Y_r^* Z^*)\|_2^2}{\|b^2\|_2^2} + \kappa \text{trace}(Y_r Y_r^*); \]

The closed forms of gradient and action of Hessian are known [HGZ14];

Their complexities respectively are

- Function evaluation: \( O(\ell \text{pns} \max_i(\log(n_i))) \);
- Gradient evaluation: \( O(\ell \text{pns} \max_i(\log(n_i))) + O(np^2) + O(p^3) \);
- Action of Hessian: \( O(\ell \text{pns} \max_i(\log(n_i))) + O(np^2) + O(p^3) \);

If \( p \ll n \) (it is true in practice), then all these complexities are dominated by \( O(\ell \text{pns} \max_i(\log(n_i))) \);
All tests are performed in Matlab R2014a on a 64 bit Ubuntu system with 3.6 GHz CPU (Intel (R) Core (TM) i7-4790).

The stopping criterion requires the norm of gradient to less than $10^{-6}$;

The number of masks $\ell$ is 6;

The coefficient $\kappa$ in the cost function is 0;

Threshold $\delta = 0.9$ for rank reduction;
Representative Riemannian Algorithms

- RNewton, RTR-Newton, LRBFGS, LRTR-SR1 and RCG;
- Noiseless measurements, i.e., $\tau = 0$;
- Initial rank $p_0 = 8$;
- Average of 10 random runs;
Representative Riemannian Algorithms

Table: $n = n_1n_2$. The subscript $\nu$ indicates a scale of $10^\nu$.

<table>
<thead>
<tr>
<th>$(n_1, n_2)$</th>
<th>$(32, 32)$</th>
<th>$(32, 64)$</th>
<th>$(64, 64)$</th>
<th>$(64, 128)$</th>
<th>$(128, 128)$</th>
<th>$(128, 256)$</th>
<th>$(256, 256)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RNewton</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$ng$</td>
<td>2.341</td>
<td>2.581</td>
<td>2.921</td>
<td>3.261</td>
<td>3.81</td>
<td>4.651</td>
<td>5.141</td>
</tr>
<tr>
<td>$f_f$</td>
<td>1.25_13</td>
<td>2.17_13</td>
<td>1.37_14</td>
<td>4.83_13</td>
<td>1.59_14</td>
<td>2.32_12</td>
<td>3.79_13</td>
</tr>
<tr>
<td>$t$</td>
<td>2.54</td>
<td>3.25</td>
<td>9.28</td>
<td>1.721</td>
<td>3.021</td>
<td>7.571</td>
<td>1.982</td>
</tr>
<tr>
<td><strong>RTR-Newton</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$nf$</td>
<td>2.821</td>
<td>2.661</td>
<td>2.841</td>
<td>3.061</td>
<td>3.421</td>
<td>3.621</td>
<td>3.361</td>
</tr>
<tr>
<td>$ng$</td>
<td>2.821</td>
<td>2.661</td>
<td>2.841</td>
<td>3.061</td>
<td>3.421</td>
<td>3.621</td>
<td>3.361</td>
</tr>
<tr>
<td>$nH$</td>
<td>3.002</td>
<td>6.132</td>
<td>4.362</td>
<td>484</td>
<td>5.272</td>
<td>5.332</td>
<td>5.772</td>
</tr>
<tr>
<td>$f_f$</td>
<td>2.11_14</td>
<td>2.86_13</td>
<td>4.53_13</td>
<td>4.69_13</td>
<td>3.03_13</td>
<td>9.74_14</td>
<td>3.45_14</td>
</tr>
<tr>
<td>$t$</td>
<td>1.83</td>
<td>5.45</td>
<td>8.23</td>
<td>1.921</td>
<td>3.421</td>
<td>6.071</td>
<td>1.422</td>
</tr>
<tr>
<td><strong>LRBFGS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$nf$</td>
<td>9.781</td>
<td>1.062</td>
<td>1.202</td>
<td>133</td>
<td>1.452</td>
<td>1.842</td>
<td>191</td>
</tr>
<tr>
<td>$ng$</td>
<td>9.61</td>
<td>1.042</td>
<td>1.162</td>
<td>1.292</td>
<td>1.402</td>
<td>1.772</td>
<td>1.872</td>
</tr>
<tr>
<td>$f_f$</td>
<td>6.40_12</td>
<td>6.82_12</td>
<td>7.61_12</td>
<td>1.04_11</td>
<td>1.58_11</td>
<td>2.24_11</td>
<td>3.55_11</td>
</tr>
<tr>
<td>$t$</td>
<td>6.00_1</td>
<td>1.03</td>
<td>1.90</td>
<td>3.37</td>
<td>6.86</td>
<td>1.591</td>
<td>3.271</td>
</tr>
<tr>
<td><strong>LRTR-SR1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$nf$</td>
<td>1.452</td>
<td>1.442</td>
<td>1.562</td>
<td>1.712</td>
<td>1.882</td>
<td>2.292</td>
<td>2.432</td>
</tr>
<tr>
<td>$ng$</td>
<td>1.452</td>
<td>1.442</td>
<td>1.562</td>
<td>1.712</td>
<td>1.882</td>
<td>2.292</td>
<td>2.432</td>
</tr>
<tr>
<td>$f_f$</td>
<td>1.24_11</td>
<td>1.08_11</td>
<td>1.50_11</td>
<td>3.67_11</td>
<td>3.10_11</td>
<td>4.82_11</td>
<td>2.04_10</td>
</tr>
<tr>
<td>$t$</td>
<td>9.97_1</td>
<td>1.64</td>
<td>3.16</td>
<td>6.20</td>
<td>1.221</td>
<td>3.141</td>
<td>6.811</td>
</tr>
<tr>
<td><strong>RCG</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$nf$</td>
<td>2.662</td>
<td>2.592</td>
<td>2.772</td>
<td>2.892</td>
<td>3.112</td>
<td>3.452</td>
<td>3.732</td>
</tr>
<tr>
<td>$ng$</td>
<td>2.552</td>
<td>250</td>
<td>266</td>
<td>2.802</td>
<td>3.022</td>
<td>3.362</td>
<td>3.652</td>
</tr>
<tr>
<td>$f_f$</td>
<td>3.11_12</td>
<td>3.47_12</td>
<td>5.42_12</td>
<td>7.94_12</td>
<td>1.16_11</td>
<td>1.60_11</td>
<td>2.53_11</td>
</tr>
<tr>
<td>$t$</td>
<td>1.18</td>
<td>2.00</td>
<td>3.74</td>
<td>7.18</td>
<td>1.471</td>
<td>3.631</td>
<td>9.541</td>
</tr>
</tbody>
</table>

LRBFGS is chosen to be the representative Riemannian algorithm.
Compare with convex programming

- FISTA [BT09] in Matlab library TFOCS [BCG11];
- $X$ can be too large to be handled by the solver;
- A low rank version of FISTA is used, denoted by LR-FISTA;
- The approach is used in [CESV13, CSV13];
- Works in practice but no theoretical results.
\begin{itemize}
\item $n_1 = n_2 = 64; \ n = n_1 n_2 = 4096;\n\item LR-FISTA stops if $\frac{\|X^{(i)} - X^{(i-1)}\|_F}{\|X^{(i)}\|_F} < 10^{-6}$ or $\text{iter} > 2000;\n\item Noise and noiseless problems are tested;\n\item For noise measurements:
  \begin{itemize}
  \item $\tau = 10^{-4}$, i.e., the signal-to-noise ratio (SNR) $10 \log_{10} \left( \frac{\|b^2\|_2^2}{\|b^2 - \hat{b}^2\|_2^2} \right)$
  \end{itemize}
  is 31.05 dB, where $b^2 = \text{diag}(Zx^* x^* Z^*)$ and $\hat{b}$ is the noise measurements;
\item Multiple $\kappa$ are used;
\end{itemize}
Table: \( k \) denotes the upper bound of the low-rank approximation in LR-FISTA. \( \# \) represents the number of iterations reach the maximum. The relative mean-square error (RMSE) is \( \min_{a:|a|=1} ||ax - x^*||_2 / ||x^*||_2 \).

<table>
<thead>
<tr>
<th>noiseless</th>
<th>Algorithm 1</th>
<th>LR-FISTA (k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>iter</td>
<td>124</td>
<td>1022 377 601 1554 2000#</td>
</tr>
<tr>
<td>nf</td>
<td>129</td>
<td>2212 804 1278 3360 4322</td>
</tr>
<tr>
<td>ng</td>
<td>124</td>
<td>1106 402 639 1680 2161</td>
</tr>
<tr>
<td>( f_f )</td>
<td>4.62 (-12)</td>
<td>8.18 (-12) 4.50 (-11) 4.64 (-12) 1.54 (-11) 1.27 (-9)</td>
</tr>
<tr>
<td>RMSE</td>
<td>6.34 (-6)</td>
<td>1.01 (-5) 1.74 (-5) 1.46 (-5) 1.10 (-4) 2.56 (-3)</td>
</tr>
<tr>
<td>( t )</td>
<td>2.12</td>
<td>1.27 (_2) 5.25 (_1) 9.35 (_1) 3.48 (_2) 6.86 (_2)</td>
</tr>
</tbody>
</table>

- Algorithm 1 is faster and gives smaller RMSE.
Table: $k$ denotes the upper bound of the low-rank approximation in LR-FISTA. RMSE denotes $\min_{a:|a|=1}\|ax - x^*\|_2/\|x^*\|_2$. $\#$ represents the number of iterations reach the maximum.

<table>
<thead>
<tr>
<th>noise</th>
<th>$\kappa$</th>
<th>Algorithm 1</th>
<th>LR-FISTA (k)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>$\text{iter}$</td>
<td>$10^{-2}$</td>
<td>84</td>
<td>409</td>
</tr>
<tr>
<td>$n_f$</td>
<td>$10^{-2}$</td>
<td>86</td>
<td>886</td>
</tr>
<tr>
<td></td>
<td>$10^{-4}$</td>
<td>129</td>
<td>2116</td>
</tr>
<tr>
<td>$n_g$</td>
<td>$10^{-2}$</td>
<td>84</td>
<td>526</td>
</tr>
<tr>
<td></td>
<td>$10^{-4}$</td>
<td>122</td>
<td>1105</td>
</tr>
<tr>
<td>$f_f$</td>
<td>$10^{-2}$</td>
<td>$1.63_{-1}$</td>
<td>$1.77_{-1}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-4}$</td>
<td>$1.80_{-3}$</td>
<td>$1.81_{-3}$</td>
</tr>
<tr>
<td>RMSE</td>
<td>$10^{-2}$</td>
<td>$1.80_{-1}$</td>
<td>$2.64_{-1}$</td>
</tr>
<tr>
<td></td>
<td>$10^{-4}$</td>
<td>$2.63_{-3}$</td>
<td>$6.46_{-3}$</td>
</tr>
<tr>
<td>$t$</td>
<td>$10^{-2}$</td>
<td>1.59</td>
<td>5.17</td>
</tr>
<tr>
<td></td>
<td>$10^{-4}$</td>
<td>2.06</td>
<td>1.21</td>
</tr>
</tbody>
</table>
Table : $k$ denotes the upper bound of the low-rank approximation in LR-FISTA. RMSE denotes $\min_{a:|a|=1} \|ax - x_*\|_2/\|x_*\|_2$. # represents the number of iterations reach the maximum.

<table>
<thead>
<tr>
<th>noise</th>
<th>$\kappa$</th>
<th>Algorithm 1</th>
<th>LR-FISTA (k) 1</th>
<th>LR-FISTA (k) 2</th>
<th>LR-FISTA (k) 4</th>
<th>LR-FISTA (k) 8</th>
<th>LR-FISTA (k) 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>iter</td>
<td>$10^{-6}$</td>
<td>128</td>
<td>1027</td>
<td>2000#</td>
<td>2000#</td>
<td>2000#</td>
<td>2000#</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>138</td>
<td>1070</td>
<td>2000#</td>
<td>2000#</td>
<td>2000#</td>
<td>2000#</td>
</tr>
<tr>
<td>nf</td>
<td>$10^{-6}$</td>
<td>132</td>
<td>2210</td>
<td>4266</td>
<td>4312</td>
<td>4336</td>
<td>4316</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>143</td>
<td>2306</td>
<td>4308</td>
<td>4322</td>
<td>4314</td>
<td>4320</td>
</tr>
<tr>
<td>ng</td>
<td>$10^{-6}$</td>
<td>128</td>
<td>1105</td>
<td>2712</td>
<td>2156</td>
<td>2168</td>
<td>2158</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>138</td>
<td>1153</td>
<td>2154</td>
<td>2161</td>
<td>2157</td>
<td>2160</td>
</tr>
<tr>
<td>$f_f$</td>
<td>$10^{-6}$</td>
<td>1.84$^{-5}$</td>
<td>1.84$^{-5}$</td>
<td>1.91$^{-5}$</td>
<td>2.35$^{-5}$</td>
<td>3.55$^{-5}$</td>
<td>7.62$^{-5}$</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>4.08$^{-7}$</td>
<td>4.08$^{-7}$</td>
<td>1.16$^{-6}$</td>
<td>6.27$^{-6}$</td>
<td>2.51$^{-5}$</td>
<td>8.89$^{-5}$</td>
</tr>
<tr>
<td>RMSE</td>
<td>$10^{-6}$</td>
<td>6.72$^{-4}$</td>
<td>6.72$^{-4}$</td>
<td>1.09$^{-3}$</td>
<td>2.10$^{-3}$</td>
<td>3.53$^{-3}$</td>
<td>6.27$^{-3}$</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>6.70$^{-4}$</td>
<td>6.70$^{-4}$</td>
<td>1.09$^{-3}$</td>
<td>2.18$^{-3}$</td>
<td>4.01$^{-3}$</td>
<td>7.29$^{-3}$</td>
</tr>
<tr>
<td>$t$</td>
<td>$10^{-6}$</td>
<td>2.13</td>
<td>1.27$^{-2}$</td>
<td>2.75$^{-2}$</td>
<td>3.01$^{-2}$</td>
<td>4.64$^{-2}$</td>
<td>7.04$^{-2}$</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2.20</td>
<td>1.34$^{-2}$</td>
<td>2.63$^{-2}$</td>
<td>2.98$^{-2}$</td>
<td>4.32$^{-2}$</td>
<td>6.91$^{-2}$</td>
</tr>
</tbody>
</table>
The Gold Ball Data

Figure: Image of the absolute value of the 256-by-256 complex-valued image. \( n = 65536 \). The pixel values correspond to the complex transmission coefficients of a collection of gold balls embedded in a medium.

Thank Stefano Marchesini at Lawrence Berkeley Notional Laboratory for providing the gold balls data set and granting permission to use it.
The Gold Ball Data

A set of binary masks contains a mask that is all 1 (which yields the original image) and several other masks comprising elements that are 0 or 1 with equal probability.

Table: RMSE and computational time (second) results with varying number and types of masks are shown in format RMSE/TIME. ♯ represents the computational time reaching 1 hour, i.e., 3.63 seconds.

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>Algorithm 1</th>
<th>LR-FISTA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>40</td>
</tr>
<tr>
<td>6 Gaussian</td>
<td>8.32&lt;sub&gt;-3&lt;/sub&gt;/4.30&lt;sub&gt;1&lt;/sub&gt;</td>
<td>8.32&lt;sub&gt;-5&lt;/sub&gt;/4.50&lt;sub&gt;1&lt;/sub&gt;</td>
</tr>
<tr>
<td>6 binary</td>
<td>7.23&lt;sub&gt;-1&lt;/sub&gt;/7.90&lt;sub&gt;2&lt;/sub&gt;</td>
<td>1.29&lt;sub&gt;-1&lt;/sub&gt;/4.24&lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
<tr>
<td>32 binary</td>
<td>2.21&lt;sub&gt;-1&lt;/sub&gt;/6.84&lt;sub&gt;2&lt;/sub&gt;</td>
<td>3.02&lt;sub&gt;-3&lt;/sub&gt;/7.36&lt;sub&gt;2&lt;/sub&gt;</td>
</tr>
</tbody>
</table>
The Gold Ball Data

Figure: Reconstructions via PhaseLift with varying number and types of masks. For the same number and types of masks, the reconstructions of

6 Gaussian masks, SNR: Inf

6 Binary masks, SNR: Inf

32 Binary masks, SNR: Inf

10 times error

10 times error

10 times error
The Gold Ball Data

- 6 Gaussian masks, SNR: 20
- 6 Binary masks, SNR: 20
- 32 Binary masks, SNR: 20

**Figure**: Reconstructions via PhaseLift with varying number and types of masks. For the same number and types of masks, the reconstructions of noisy measurements with SNR 20 are shown. The error is shown in the images by magnifying 10 times.
Conclusion

- A low-rank problem is proposed to replace optimization problems on Hermitian positive semidefinite matrices;
- The first order optimality condition is given;
- For the PhaseLift problem, an algorithm based on a rank reduce strategy and a state-of-the-art Riemannian algorithm is suggested;
- Experiments of noise, noiseless, Gaussian masks and binary masks are tested and show that the new algorithm is more efficient and effective than the LR-FISTA algorithm.
P.-A. Absil, R. Mahony, and R. Sepulchre.  
*Optimization algorithms on matrix manifolds.*  

C. G. Baker.  
*Riemannian manifold trust-region methods with applications to eigenproblems.*  

S. Becker, E. J. Cand, and M. Grant.  
Templates for convex cone problems with applications to sparse signal recovery.  
Diffractive imaging for periodic samples: retrieving one-dimensional concentration profiles across microfluidic channels.  

S. Burer and R. D. C. Monteiro.  
A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization.  

Y. M. Bruck and L. G. Sodin.  
On the ambiguity of the image reconstruction problem.  


PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming. 

W. Huang, P.-A. Absil, and K. A. Gallivan.
A Riemannian symmetric rank-one trust-region method. 
*Mathematical Programming*, February 2014. 

R. W. Harrison.
Phase problem in crystallography. 
M. H. Hayes.  
The reconstruction of a multidimensional sequence from the phase or magnitude of its fourier transform.  
_ElEEE Transactions on Acoustics speech and signal processing, 30(2):140–154, 1982._

W. Huang, K. A. Gallivan, and P.-A. Absil.  
A Broyden class of quasi-Newton methods for Riemannian optimization.  
_Submitted for publication, 2014._

Wen Huang, K. A. Gallivan, and X Zhang.  
Solving Phaselift by low-rank Riemannian optimization methods for complex semidefinite constraints.  
_Submitted for publication, 2014._

Low-rank optimization on the cone of positive semidefinite matrices.  
Extending X-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes. 

W. Ring and B. Wirth. 
Optimization methods on Riemannian manifolds and their application to shape space. 
doi:10.1137/11082885X.

J. L. C. Sanz. 
Mathematical considerations for the problem of Fourier transform phase retrieval from magnitude. 

